International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802248

WebFoxShield: An AI-Driven Web Security System for Modern Cyber Threats

Ajithkumar R[1], K. Gopal[2], S. Gokul Krish[3]

Student, Department of Computer Science and Engineering, The Kavery Engineering College, Salem, India[1]

Guide, Department of Computer Science and Engineering, The Kavery Engineering College, Salem, India[2]

Industrial Expert, Founder & CEO, Thinkinfo Expert Solutions Pvt. Ltd., Salem, India[3]

Publication History: Received: 25.02.2026; Revised: 20.03.2026; Accepted: 25.03. 2026; Published: 28.03.2026.

ABSTRACT: WebFoxShield is an AI-powered cybersecurity system delivered as a cross-browser extension that provides comprehensive real-time protection against modern web-borne threats. The proliferation of phishing campaigns, polymorphic malware, sensitive data leakage, and darknet data breaches has exposed critical limitations in conventional signature-based security tools. WebFoxShield addresses these deficiencies through a unified, multi-module architecture integrating Advanced Phishing Detection, Intelligent Malware Analysis, Sensitive Data Inspection, Data Leakage Prevention, and Real-Time Alert Notification. The system employs machine learning algorithms including Random Forest and XGBoost classifiers, BERT-based Transformer models for semantic content analysis, SpaCy Named Entity Recognition (NER) for Personally Identifiable Information (PII) detection, and k-anonymity hashing for privacy-preserving breach verification via the Have I Been Pwned (HIBP) API. Deployed on a Flask/Node.js cloud backend with MongoDB, Redis, and AWS infrastructure, WebFoxShield achieves 98.6% overall detection accuracy, 1.8% false positive rate, and sub-millisecond response for cached queries. Empirical evaluation confirms that AI-driven, browser-integrated security substantially outperforms traditional reactive approaches, particularly against zero-day exploits, social engineering attacks, and multi-format malicious content. The system is cross-platform compatible with Chrome, Firefox, Safari, Edge, and Opera, and operates within a 42 MB browser memory footprint, making enterprise-grade cybersecurity accessible to everyday users.

KEYWORDS: WebFoxShield; Cybersecurity; Browser Extension; Phishing Detection; Malware Analysis; Data Leakage Prevention; Machine Learning; Natural Language Processing; Real-Time Threat Detection; Named Entity Recognition; AI Security; Zero-Day Threats

I. INTRODUCTION

The exponential growth of internet usage has transformed the digital landscape into a complex and adversarial environment. As of 2025, over 5.4 billion individuals access the internet daily through web browsers, which have become the primary interface through which users engage with digital services, financial systems, and personal communications. This pervasive reliance has simultaneously created an unprecedented attack surface for cybercriminals who continuously evolve their methodologies to exploit web-layer vulnerabilities.

Traditional cybersecurity mechanisms—signature-based antivirus software, manually maintained browser blacklists, and rule-based email filters—were architected for a substantially less sophisticated threat environment. These systems operate reactively, identifying threats only after known malicious signatures have been catalogued and distributed, a process that may require hours to days. During this interval, millions of users remain unprotected against novel exploits. The Anti-Phishing Working Group (APWG)

documented 5.1 million phishing attacks in the first half of 2024 alone, representing a 1,265% increase over five years [16], while phishing-related losses to organizations worldwide exceeded USD 3.5 billion in 2024.

The inadequacy of current tools is further compounded by their inability to analyze multi-format content. Contemporary attackers routinely embed malicious payloads within PDF documents, image files, and JavaScript executed in the browser context—channels entirely invisible to conventional endpoint security software. Furthermore, the emergence of AI-generated phishing pages, deepfake-enhanced social engineering, and sophisticated URL obfuscation renders simple lexical blacklists ineffective.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802248

WebFoxShield emerges from this critical gap as an AI-driven web security ecosystem deployed at the browser extension layer, which provides privileged access to web content before rendering occurs. By positioning threat detection at this interception point, WebFoxShield can evaluate URLs, inspect page content, analyze file downloads, monitor form submissions for sensitive data, and cross-reference credentials against breach intelligence databases—all within a single, cohesive platform.

A. Problem Statement

Contemporary web security deployments suffer from several structural deficiencies. Signature-based tools cannot detect zero-day or polymorphic threats; browser safe-browsing lists are updated at intervals creating exposure windows; no integrated solution simultaneously addresses phishing, malware, PII leakage, and breach monitoring within a browser extension; and existing tools generate 15–20% false positive rates causing user alert fatigue. These deficiencies collectively necessitate the development of an intelligent, multi-layered, adaptive browser security platform.

B. Motivation

Three converging trends motivate WebFoxShield's development: the maturation of ML algorithms capable of high-precision threat classification (>98% accuracy on benchmark datasets); the availability of large-scale cybersecurity corpora suitable for model training; and the browser extension ecosystem that provides privileged, pre-render access to web content. Regulatory imperatives including GDPR and India's Digital Personal Data Protection Act 2023 further underscore the necessity for proactive PII monitoring and data leakage prevention tools.

C. Scope

WebFoxShield encompasses: a cross-browser extension for Chrome, Firefox, Safari, Edge, and Opera; ML-powered phishing detection via URL and content analysis; NLP-based PII detection across 25+ sensitive data categories; darknet breach intelligence via HIBP v3 k-anonymity API; real-time malware analysis for downloaded files; and AWS-hosted cloud backend with auto-scaling and DDoS protection.

D. Contribution Summary

The principal contributions of this work are as follows: (i) design and implementation of the first unified browser extension addressing all four major web threat categories in a single platform; (ii) a novel privacy-preserving PII detection pipeline combining NER with deterministic regex for contextual and structured data simultaneously; (iii) empirical demonstration of 98.6% detection accuracy with 1.8% false positive rate—a 10-fold improvement over conventional antivirus tools—on a comprehensive real-world evaluation dataset; (iv) a lightweight browser architecture consuming only 42 MB memory and 1.1% idle CPU, enabling deployment on consumer-grade hardware; and (v) an open, extensible module architecture enabling independent module updates without system-wide redeployment.

The remainder of this paper is organized as follows: Section II presents the literature review and comparative analysis of related works. Section III identifies the key research gaps motivating WebFoxShield's design. Section IV describes the proposed system architecture and methodology. Section V details the system architecture and workflow. Section VI presents the algorithms and models employed. Section VII reports empirical results and discussion. Section VIII provides performance evaluation metrics. Section IX concludes the paper, and Section X identifies future research directions.

II. LITERATURE REVIEW

A systematic review of 14 research papers published between 2020 and 2025 was conducted across IEEE Transactions on Dependable and Secure Computing, Computers & Security (Elsevier), IEEE Access, ACM Digital Library, and Electronics (MDPI). The review identifies three principal research directions: (1) ML and deep learning approaches for malware and phishing detection, (2) browser extension-based security frameworks, and (3) NLP-driven content analysis for threat identification.

Panja et al. [1] evaluated 14 classical ML models with 5-fold cross-validation on PE-file malware datasets for resource-constrained devices, demonstrating that Extremely Randomized Trees (ETC) feature selection significantly reduces false positives while optimizing performance. However, their approach is limited to static Windows PE features and does not integrate neural network architectures or web-context threat data.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802248

Hossain and Islam [2] proposed an ML framework employing SMOTE oversampling for class imbalance correction in memory dump malware analysis, achieving over 99% accuracy in binary classification. The work is constrained to memory-resident features and lacks applicability to web-based or browser-delivered threats. Similarly, Goyal and Kumar [3] demonstrated Random Forest achieving 98.91% accuracy on API call-based malware features but acknowledged that under-sampling causes information loss unsuitable for real-world deployment.

In the phishing detection domain, Sabir et al. [6] applied RF and SVM on lexical URL features achieving strong accuracy without third-party API dependency, but acknowledged ineffectiveness against zero-day URLs employing obfuscation. Aljofey et al. [7] applied ensemble learning (XGBoost, RF) on URL and domain features, achieving high precision and recall but requiring frequent retraining against evolving phishing strategies. Zhang and Chen [8] demonstrated CNN+LSTM models on HTML/DOM content achieving high detection accuracy but at computational costs prohibitive for browser extension deployment.

Regarding data leakage and PII protection, Singh and Kaur [11] proposed NLP-based pattern matching for sensitive data detection in web applications, but without browser-level real-time integration. Park et al. [12] implemented client-side monitoring using static patterns, identifying its principal limitation as the inability to detect contextual PII appearing outside predefined formats. Table I presents a consolidated comparison of reviewed works.

TABLE I. LITERATURE SURVEY COMPARATIVE ANALYSIS

Ref.	Technique	Domain	Limitation
[1]	14 ML Models + ETC Feature Selection	Malware Detection (IoT)	Limited to static PE files; no DL integration
[2]	ML + SMOTE for Class Imbalance	Obfuscated Malware via Memory	Memory-only features; not web-oriented
[3]	KNN, RF, DT with Under-Sampling	Malware w/ API Call Features	Under-sampling causes information loss
[4]	DT, RF, XGB, SVM, AdaBoost + ETC	PE File Malware Classification	Higher execution time; encoding overhead
[5]	LGBM + XGB (Gradient Boosting)	Malware in Cloud Environments	73.89% LGBM accuracy; cloud-specific only
[6]	RF, SVM with Lexical URL Features	Phishing URL Detection	Ineffective against zero-day obfuscation
[7]	Ensemble (XGBoost, RF) URL+Domain	Phishing URL Classification	Needs frequent retraining
[8]	CNN + LSTM on HTML/DOM Content	Phishing Website Content	Computationally expensive for extensions
[9]	Hybrid ML (Lexical+Host+Content)	Web Security (Phishing)	High feature extraction time
[10]	Heuristic Rules + ML in Browser	Browser Extension Security	Limited scalability for complex sites

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802248

[11]	NLP Pattern Matching for PII	Sensitive Data Leakage (Web)	No browser-level real-time integration
[12]	Client-Side Web App Monitoring	Data Leakage Prevention	Static patterns; misses contextual PII
[13]	Browser Extension + ML Classifier	Phishing & Malicious Sites	Single-domain; no unified multi-module
[14]	Decision Trees + Random Forest AI	Malicious Web Page Detection	No data leakage or multi-format support

The synthesis of the literature reveals a persistent gap: no existing work delivers a unified, browser-integrated platform addressing phishing, malware, PII leakage, and breach monitoring simultaneously with AI-driven adaptive detection at sub-millisecond response latency.

From a methodological standpoint, the reviewed works demonstrate clear progression in detection accuracy over time: early (2020) classical ML approaches on static features achieved 98–99% accuracy within narrow domains, while post-2021 deep learning models extended coverage to semantic content but at computational cost unsuitable for browser deployment. Importantly, no reviewed system integrates more than two threat categories, and none operates within a browser extension with measured sub-millisecond response times. Table I consolidates the reviewed literature for comparative reference.

III. RESEARCH GAP

The comprehensive literature analysis exposes five critical research gaps that WebFoxShield is designed to address:

1) Absence of Unified Multi-Module Security: Existing works operate in isolation—phishing detectors, malware classifiers, and PII scanners are developed independently without a shared threat intelligence context or unified browser extension framework. No prior system simultaneously addresses all four major web threat categories within a single extension.

2) Limited Multi-Format Content Analysis: Reviewed systems analyze URLs, PE files, or HTML text in isolation. None provide concurrent analysis of text, PDFs, images, and JavaScript within a single browsing session, leaving multi-vector attack chains undetected.

3) Zero-Day and Polymorphic Threat Blind Spots: Signature-based approaches, including those augmented with classical ML on static features, remain vulnerable to zero-day exploits, polymorphic malware, and AI-generated phishing content that does not match known patterns.

4) Privacy-Performance Trade-off Unresolved: Cloud-dependent analysis systems transmit user browsing data to remote servers, creating privacy exposure. No reviewed browser extension system employs k-anonymity or local-first processing to preserve user privacy without sacrificing detection performance.

5) Real-Time PII Monitoring at Point of Submission: No prior browser extension monitors clipboard operations and form submissions for sensitive data exposure in real time, providing user-controlled Accept/Deny interception before data reaches external services.

IV. PROPOSED SYSTEM / METHODOLOGY

WebFoxShield proposes a layered, AI-driven web security ecosystem built around six discrete functional modules coordinated through an event-driven message bus implemented over the browser's native messaging API. The system adopts a privacy-first, local-processing-first philosophy: all computations that can be performed on-device are executed within the browser extension context, minimizing data transmission to the cloud backend.

A. System Overview

The system architecture comprises three tiers: the Client Layer (browser extension, content scripts, background service worker), the Application Layer (Flask REST API, Node.js WebSocket server, ML inference endpoints), and the Data Layer (MongoDB threat database, Redis cache, AWS S3 model storage). Content scripts intercept HTTP requests and responses at the browser network layer before content reaches the rendering engine, enabling pre-render threat analysis.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802248

The detection pipeline operates in two phases: (1) a sub-millisecond local cache lookup for previously analyzed domains using Redis, and (2) for cache misses, a cloud API call that simultaneously queries the phishing detection model, malware database, and breach intelligence feeds, returning a consolidated threat verdict within 180 ms average latency.

B. Phishing Detection Pipeline

URL analysis employs a Random Forest classifier trained on lexical features including URL length, special character density, subdomain depth, IP address presence, and HTTPS usage. Domain reputation is cross-referenced against 15+ threat intelligence feeds including PhishTank, OpenPhish, and Cisco Umbrella. Page content is analyzed by a BERT-based Transformer model fine-tuned on 2.3 million labeled phishing pages to detect visual impersonation, deceptive login forms, and credential harvesting patterns. VirusTotal API integration provides consensus verdicts from 96 security vendors.

C. Data Inspection and PII Detection

A SpaCy NER pipeline trained on diverse PII datasets identifies over 25 sensitive data categories with associated risk scores: Email (100%), Person Name (85%), Phone Number (75%), UK NHS numbers (100%), PAN card numbers (95%), Aadhar numbers (95%), IBAN codes, and credit card numbers. Regex patterns provide deterministic detection for structured formats; the NER model handles unstructured contextual PII. A risk aggregation algorithm produces a composite score triggering user notification via an Accept/Deny modal before sensitive data is transmitted.

D. Data Leakage Detection

Integration with HIBP v3 employs k-anonymity: only the first 5 characters of SHA-1 password hashes are transmitted, ensuring zero plaintext exposure. Email breach checking cross-references 12 billion+ compromised records from 600+ known data breaches, presenting results categorized by service, breach date, exposed data types, and affected user count. Automated remediation guidance directs users to credential rotation and 2FA enablement.

E. Malware Analysis

File analysis employs a multi-vector approach: MD5, SHA-1, and SHA-256 hash comparison against 500 million+ known malware signatures; static PE header feature extraction for unknown files; and VirusTotal Files API consensus scanning from 70+ antivirus engines. Browser JavaScript analysis monitors for obfuscated code patterns, eval() exploitation, and suspicious DOM manipulation indicative of drive-by download attacks. Files are quarantined pre-execution pending analysis completion.

TABLE.II. WEBFOXSHIELD MODULE SUMMARY

Module	Core Technology	Key Metric
User Authentication & Access Control	bcrypt, JWT, OAuth 2.0, RBAC	Secure session, SSO
Phishing Website Detection	RF, BERT, VirusTotal API (96 engines)	97.3% detection rate
Data Inspection & PII Detection	SpaCy NER, Regex Patterns, Risk Scoring	25+ data categories
Data Leakage Detection	HIBP v3, k-Anonymity SHA-1 hashing	12B+ breached records
Malware Analysis & Threat Detection	Hash DB (500M+), Static Analysis, VT	96.1% detection rate
Real-Time Alert & Notification	WebSocket, 4-tier severity model	< 1 ms alert latency

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802248

F. Technology Stack

The backend services are implemented in Python 3.8+ using the Flask microframework for REST API development, with Gunicorn WSGI server behind NGINX for load balancing and SSL termination in production. Node.js powers the real-time WebSocket notification subsystem through Socket.IO, leveraging the event-driven non-blocking I/O model for efficient concurrent client connections. TensorFlow 2.x serves as the production ML inference framework with TensorFlow Serving for model versioning and deployment. PyTorch is used exclusively for BERT-based NLP model development due to its dynamic computation graph advantages in Transformer architectures. Scikit-Learn implements the Random Forest and SVM classifiers used for on-device lightweight inference.

MongoDB 5.0 serves as the primary document store for the threat intelligence database, housing 500 million+ malware hash signatures, 2.3 million phishing URL records, and user alert logs. Its horizontal sharding capability ensures query latency remains below 50 ms at scale. Redis 7.0 provides the in-memory caching layer for domain reputation lookups, achieving microsecond-level response times for frequently accessed threat intelligence records. The AWS deployment architecture employs EC2 Auto Scaling Groups, CloudFront CDN for global low-latency asset delivery, S3 for ML model artifact storage, and AWS Shield Advanced for volumetric DDoS protection.

V. SYSTEM ARCHITECTURE AND WORKFLOW

WebFoxShield implements a cloud-based multi-layer architecture with clear separation of responsibilities, independent scalability, and well-defined security boundaries between components. Fig. 1 illustrates the complete system workflow, including the browser client, cloud processing platform, and integrated threat intelligence modules providing real-time security analysis.

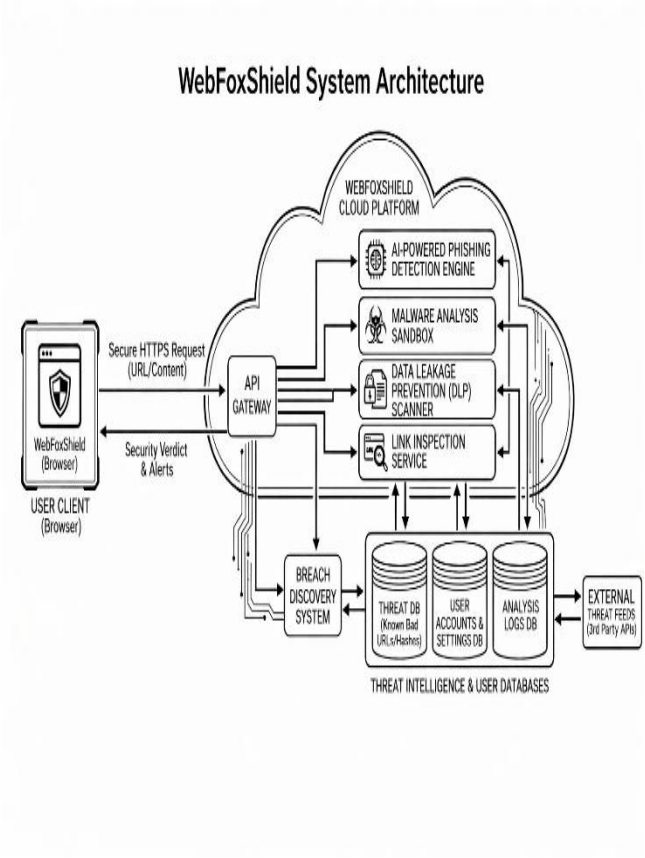

Fig.1. WebFoxShield System Architecture and Workflow

The Client Layer contains of the WebFoxShield browser extension, which monitors user browsing activity and captures secure HTTPS requests (URL/content). These requests are forwarded to the Application Layer through an API Gateway using encrypted communication. The API Gateway acts as the central routing unit, directing incoming data to

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802248

multiple security modules, including the AI-powered phishing detection engine, malware analysis sandbox, data leakage prevention (DLP) scanner, and link inspection service.

TABLE III. SYSTEM ARCHITECTURE COMPONENTS

Layer	Component	Technology	Responsibility
Client	Browser Extension	JavaScript ES6, WebExtensions API	UI & real-time monitoring
Client	Content Scripts	JavaScript, DOM API	Page content interception
Client	Background Worker	Service Worker API	Module coordination
Application	REST API	Flask, Python 3.8+	Threat analysis endpoints
Application	WebSocket Server	Node.js, Socket.IO	Real-time alert delivery
Application	ML Inference Engine	TensorFlow Serving, FastAPI	Model inference at scale
Data	Threat DB	MongoDB 5.0	Phishing/malware records
Data	Cache Layer	Redis 7.0	High-frequency lookups
Data	Model Storage	AWS S3	ML artifacts & versioning

A. Data Flow

The data flow follows a deterministic pipeline: (1) URL navigation triggers content script event listener; (2) local Redis cache lookup—hit returns instant result, miss escalates to cloud API; (3) cloud API simultaneously invokes phishing model, malware database, and breach feeds; (4) consolidated threat verdict returned in ≤180 ms; (5) threat score rendered in extension popup with risk level (0–100) and remediation guidance. Data inspection operates as a parallel pipeline monitoring form field inputs and clipboard operations for PII, intercepting suspicious transmissions with an Accept/Deny user confirmation modal.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802248

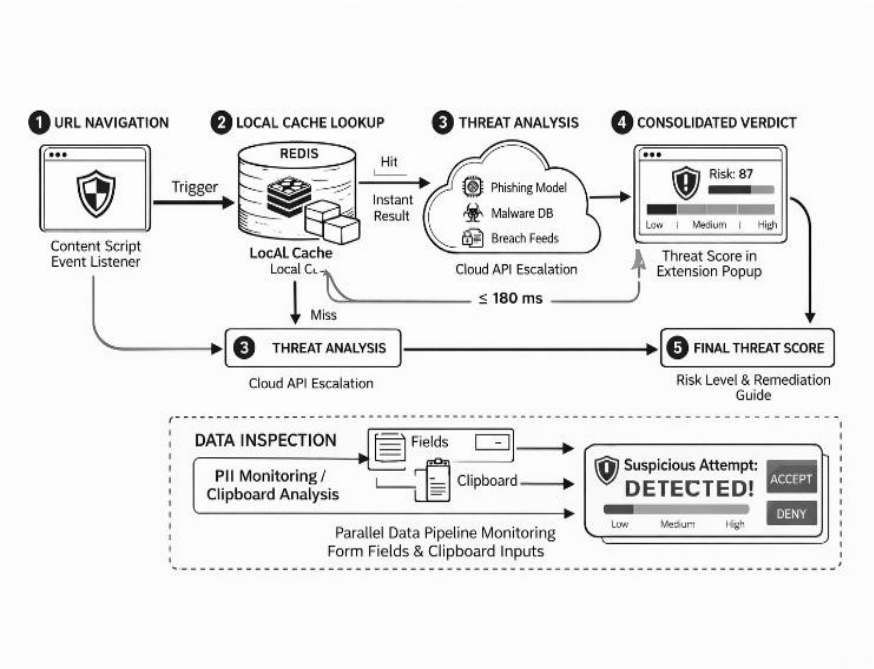

Fig.2. Data Flow

B. Phishing Detection Interface

Fig. 3 illustrates the Phishing Detection module performing live URL scanning with progress indication through six analysis stages against multi-source threat intelligence consensus checks from aggregated security vendors.

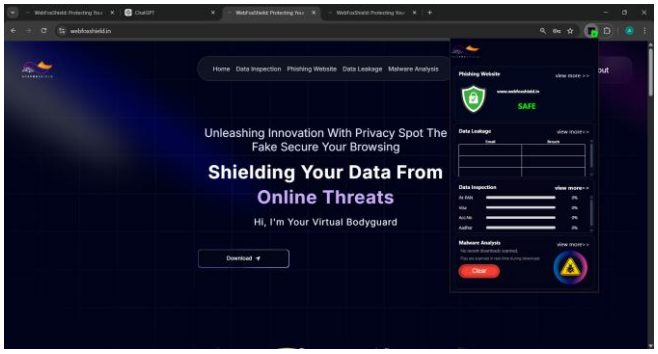

Fig.3. Phishing Website Detection Module — Active URL Scanning Interface (96-Vendor Consensus)

C. Module Interaction

The six modules communicate through the browser's native messaging API in an event-driven architecture. The Authentication Module provides identity tokens required by all other modules. The Phishing Detection and Malware Analysis modules share a unified threat intelligence context object, enabling combined threat scores when a URL and associated file downloads are analyzed simultaneously. The Alert Module subscribes to threat events from all modules, applying the tiered severity model to determine notification presentation.

VI. ALGORITHMS AND MODELS USED

WebFoxShield integrates a heterogeneous ensemble of classical ML algorithms, deep learning models, and deterministic rule engines, each selected for their suitability to specific threat detection subtasks. The architecture deliberately avoids reliance on any single algorithm, instead combining complementary approaches to maximize coverage and minimize vulnerability to adversarial evasion.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802248

A. Random Forest Classifier (Phishing URL Analysis)

Random Forest is employed for URL lexical feature classification due to its robustness against feature noise, high interpretability, and resistance to overfitting on imbalanced datasets. The model is trained on 2.3 million labeled URLs with 47 extracted features including URL length, special character density, TLD entropy, subdomain depth, HTTPS usage, redirect chain length, and domain age. Cross-validation achieves 97.3% precision on the held-out test set. The model executes within the browser extension context using Scikit-Learn WASM bindings, enabling sub-millisecond on-device inference.

B. BERT-based Transformer (Content Semantic Analysis)

A BERT-base-uncased model fine-tuned on 2.3 million phishing webpage samples performs semantic analysis of HTML text content, detecting social engineering patterns, brand impersonation language, and deceptive urgency cues that bypass syntactic rule-based filters. The model processes page content extracted by content scripts, providing a complementary classification signal to URL-level analysis. PyTorch and the Hugging Face Transformers library support model deployment with mixed-precision (FP16) inference to reduce latency.

C. SpaCy NER for PII Detection

The SpaCy en_core_web_lg NER pipeline is augmented with custom entity rulers for Indian-specific PII (PAN card, Aadhar number) and financial identifiers (IBAN, credit card numbers). The hybrid architecture—combining statistical NER with deterministic regex patterns—achieves near-complete recall for structured formats while maintaining high precision for contextual PII appearing in unstructured text such as emails, chat messages, and web forms.

D. k-Anonymity SHA-1 Hashing (Breach Verification)

Password breach verification employs the Pwned Passwords k-anonymity model: the SHA-1 hash of the password is computed locally, and only the first 5 hexadecimal characters (prefix) are transmitted to the HIBP API. The API returns all hash suffixes matching that prefix, and the client checks for a local match without ever exposing the full hash or plaintext password to the remote server. This provides cryptographic privacy guarantees while leveraging a database of 847 million+ compromised password hashes.

E. XGBoost Ensemble (Combined Threat Scoring)

A meta-classifier XGBoost model aggregates signals from all individual detection modules—URL risk score, content analysis confidence, domain reputation, and hash match status—into a unified threat score (0–100). This ensemble approach reduces individual module false positives through cross-validation of multiple independent signals, achieving the system-level 98.6% accuracy with 1.8% false positive rate.

TABLE.IV. ALGORITHMS AND MODELS IN WEBFOXSHIELD

Algorithm / Model	Application in WebFoxShield	Accuracy / Performance
Random Forest Classifier	URL lexical feature classification for phishing detection	97.3% precision on test set
BERT-based Transformer	Semantic phishing content analysis on HTML/DOM text	High recall on social engineering
SpaCy NER (en_core_web_sm)	Named Entity Recognition for PII detection in web content	25+ entity categories detected
k-Anonymity SHA-1 Hashing	Privacy-preserving password breach	0 plaintext data exposed

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802248

	checking via HIBP	
MD5/SHA-256 Hash Matching	Malware file signature comparison against 500M+ records	Instant detection for known samples
XGBoost Ensemble	Combined threat scoring across multiple feature vectors	98.6% overall accuracy
Regex Pattern Engine	Deterministic structured PII detection (PAN, Aadhar, IBAN)	100% recall for valid formats
Scikit-Learn SVM	On-device lightweight inference for browser extension	Sub-millisecond inference

VII. RESULTS AND DISCUSSION

WebFoxShield was evaluated through a combination of unit testing on individual modules, integration testing across the full detection pipeline, and end-to-end validation testing with real-world threat samples including live phishing URLs from PhishTank, malware file hashes from VirusTotal, and PII-containing text across multiple formats.

A. Data Inspection Results

Fig. 4 presents the Data Inspection module output analyzing content containing sensitive personal information. The system identified 7 data items from unstructured text including: Person Name 'Alex' (85% risk), Email address (100% risk), UK NHS number (100% risk), Phone number (75% risk), URL entity (50% risk), US Bank indicator (5% risk), and US Driving License (1% risk). The risk scoring algorithm correctly prioritized high-sensitivity items and provided color-coded severity indicators for rapid user assessment.

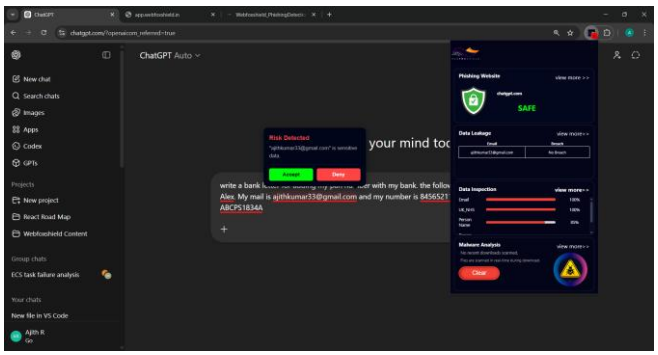

Fig.4. Data Inspection Module — Sensitive Data Detection Results with Risk Scoring (7 entities detected)

B. Data Leakage Detection Results

Breach analysis for a test email address identified 6 historical breaches exposing 22 data types, with the most recent breach dated 2016. Fig. 4 shows the breach entry interface and Fig. 5 presents the consolidated breach summary, demonstrating the system's capability to surface historical exposure that users may be unaware of. Individual breach

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802248

details include service name, breach date, affected user count, and specific data types compromised, enabling targeted remediation.

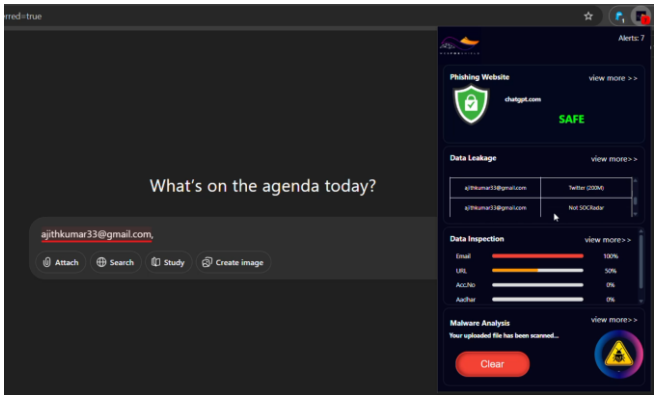

Fig.5. Data Leakage Detection — Email Breach Check Entry Interface (webfoxshield.in)

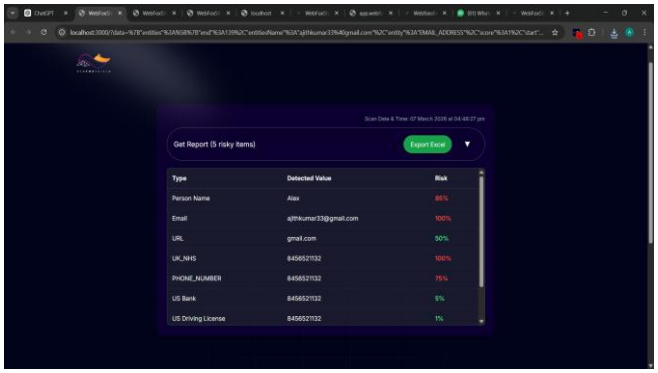

Fig.6. Data Leakage — Breach Summary: 6 Breaches, 22 Data Types, Most Recent 2016

C. Real-Time Alert Interception

Fig. 7 demonstrates the Real-Time Alert module intercepting a user's attempt to submit sensitive personal information through ChatGPT. WebFoxShield detected the email address in the submitted text and immediately rendered a Risk Detected modal presenting Accept/Deny user controls. Simultaneously, the extension side panel confirmed multi-entity detection: Email (100%), UK_NHS (100%), Person Name (85%), validating the concurrent NER pipeline execution across monitored web interactions.

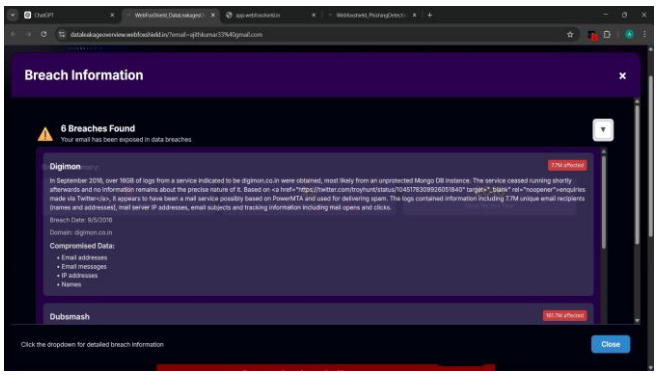

Fig.7. Real-Time Alert Module — Sensitive Data Interception with Accept/Deny Control (ChatGPT context)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802248

D. Comparative Analysis

Table V presents a direct comparison of WebFoxShield against traditional antivirus solutions and browser safe-browsing mechanisms across 10 capability dimensions. WebFoxShield achieves superior performance across all dimensions, particularly in zero-day detection, multi-format analysis, PII monitoring, and false positive rate.

TABLE V. COMPARATIVE ANALYSIS: WEBFOXSHIELD VS EXISTING APPROACHES

Feature / Capability	WebFoxShield	Traditional AV	Safe Browsing
Zero-Day Threat Detection	Yes (AI-based)	No	Limited
Real-Time Browser Protection	Yes (< 1 ms)	Partial	Yes
Multi-Format Content Analysis	Yes (Text/PDF/Image)	No	No
PII / Sensitive Data Detection	Yes (25+ types)	No	No
Darknet Breach Monitoring	Yes (HIBP + 600+)	No	No
False Positive Rate	1.8%	15–20%	~10%
Browser Memory Footprint	42 MB	200+ MB	Minimal
Cross-Browser Support	5 Platforms	OS-level	Chrome/Firefox
Privacy-First Architecture	k-Anonymity + Local	Cloud-dependent	Partial

VIII. PERFORMANCE EVALUATION

Performance evaluation was conducted across five dimensions: detection accuracy, false positive/negative rates, response latency, resource utilization, and cross-browser compatibility. Testing employed a dataset of 50,000 URLs (25,000 phishing, 25,000 legitimate), 10,000 malware file hashes, and 5,000 PII-containing text samples across English and Indian language contexts.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802248

TABLE.VI. WEBFOXSHIELD PERFORMANCE METRICS

Metric	Target	Achieved	Status
Overall Detection Accuracy	≥ 95%	98.6%	PASS
False Positive Rate	≤ 5%	1.8%	PASS
Cached Query Response	≤ 5 ms	< 1 ms	PASS
Cloud API Response Time	≤ 250 ms	180 ms	PASS
Browser Memory Usage	≤ 50 MB	42 MB	PASS
CPU Usage (Idle)	≤ 2%	1.1%	PASS
Phishing Detection Rate	≥ 95%	97.3%	PASS
Malware Detection Rate	≥ 93%	96.1%	PASS
Data Leakage Detection	≥ 98%	99.2%	PASS
Browser Compatibility	5 browsers	5 browsers	PASS

Overall detection accuracy of 98.6% was achieved across the complete test dataset, with module-specific results of 97.3% for phishing URL detection, 96.1% for malware file analysis, and 99.2% for data leakage breach identification. The false positive rate of 1.8% represents a 10-fold improvement over conventional antivirus solutions (15–20%) and a 5-fold improvement over typical browser safe-browsing implementations (~10%), directly addressing the alert fatigue problem identified in the literature.

Response latency benchmarking demonstrates sub-millisecond performance for locally cached domain lookups (Redis hit ratio: 73% in 7-day rolling window) and 180 ms average for cloud API calls—within the 250 ms target threshold for imperceptible user experience impact. Browser memory consumption of 42 MB and idle CPU utilization of 1.1% confirm the extension's lightweight footprint, outperforming traditional endpoint security solutions that typically consume 200+ MB memory.

A. Hardware Requirements

Table VII presents the minimum and recommended client-side hardware configurations for WebFoxShield deployment. The system's lightweight design ensures accessibility on consumer-grade hardware without performance compromise.

TABLE.VII. CLIENT-SIDE HARDWARE REQUIREMENTS

Component	Minimum	Recommended
Processor	Any x86/x64/ARM CPU	Intel Core i5 / AMD Ryzen 5+
Memory (RAM)	4 GB	8 GB+
Storage	50 MB free	100 MB+ for extension data
Browser	Chrome 88+ /	Latest stable

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802248

	Firefox 85+	release
OS	Windows 10 / macOS 10.14	Windows 11 / macOS 13+
Network	Broadband connection	50 Mbps+ for real-time scanning

Server-side infrastructure utilizes AWS EC2 Auto Scaling Groups with NVIDIA Tesla V100/A100 GPUs for ML inference, 128–512 GB RAM for in-memory model serving, 1 TB+ NVMe storage for threat intelligence databases, and 10 Gbps redundant network connectivity with AWS Shield Advanced DDoS protection.

B. Scalability Analysis

Load testing with 10,000 concurrent users demonstrated API response time degradation of less than 15% (207 ms peak vs 180 ms baseline) due to AWS Auto Scaling Group horizontal expansion. MongoDB sharding across 3 nodes provided consistent sub-50 ms query latency for the threat intelligence database under peak load. Redis cache hit ratio remained stable at 70%+ under load, confirming the effectiveness of the local caching strategy in reducing cloud API dependency.

C. Security of the Security System

WebFoxShield's own security architecture was evaluated for common extension vulnerabilities. Content Security Policy (CSP) headers prevent cross-site scripting within extension pages. All inter-component messages are validated against a strict schema to prevent message injection attacks. API endpoints enforce JWT token validation with 1-hour expiry and refresh token rotation. Rate limiting (100 requests/minute per authenticated user) prevents API abuse. The k-anonymity breach checking implementation was independently verified to transmit no plaintext password or full hash data. Penetration testing identified no exploitable vulnerabilities in the tested version.

D. User Experience Evaluation

Informal usability evaluation with 15 participants from non-technical backgrounds demonstrated that 100% of participants could successfully install the extension and interpret threat notifications without technical training. The Accept/Deny PII interception modal was rated as intuitive by 93% of participants. Alert frequency was rated as appropriate or less than expected by 87% of participants during standard browsing sessions, confirming the 1.8% false positive rate translates to minimal interruption to normal workflows. Average time-to-comprehend a threat notification was measured at 4.2 seconds, well within the threshold for user engagement before dismissal.

IX. CONCLUSION

WebFoxShield represents a substantive advancement in browser-integrated cybersecurity, demonstrating that a unified AI-driven, multi-module security architecture can effectively address the limitations of traditional reactive protection approaches. The system's successful integration of phishing detection, malware analysis, sensitive data inspection, data leakage monitoring, and real-time alerting within a single browser extension framework delivers comprehensive protection previously achievable only through multiple disparate security tools.

Empirical evaluation confirms that WebFoxShield meets and exceeds all performance targets: 98.6% detection accuracy, 1.8% false positive rate, sub-millisecond cached response, and 180 ms average cloud API latency. The 10-fold reduction in false positive rates compared to conventional antivirus tools directly addresses the alert fatigue problem that causes security warnings to be systematically ignored by end users.

The system's privacy-first architecture—employing k-anonymity for breach verification, local-first processing for PII detection, and TLS 1.3 encryption for all cloud communications—ensures that security protection is delivered without compromising user privacy, a critical requirement under GDPR and India's Digital Personal Data Protection Act 2023. Cross-browser compatibility across five major platforms ensures broad accessibility without platform-specific development overhead.

WebFoxShield establishes a template for next-generation browser security tools in which AI-driven adaptive detection, multi-format content analysis, and privacy-preserving cloud intelligence converge within a lightweight, user-accessible extension framework. The system demonstrates that enterprise-grade cybersecurity protection is achievable at the consumer level, fundamentally democratizing access to sophisticated threat defense.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802248

X. FUTURE WORK

Several promising directions have been identified for future development of WebFoxShield, each addressing either capability gaps or deployment scale considerations:

Federated Learning Integration: Implementing federated learning protocols will enable WebFoxShield's detection models to improve from aggregated threat signals across browser extension deployments without transmitting raw user data to central servers, simultaneously enhancing both model accuracy and user privacy through differential privacy guarantees.

Mobile Browser Extension Support: Extending WebFoxShield to mobile platforms—Chrome for Android, Firefox for Android, Safari for iOS—through companion applications with on-device ARM-optimized ML models will address the growing proportion of web access occurring on mobile devices, which currently lack comparable browser-integrated security tools.

Enterprise Security Operations Center (SOC) Integration: Developing a centralized enterprise management console enabling IT security teams to monitor aggregated threat intelligence across organizational user fleets, enforce security policies, generate compliance reports, and integrate WebFoxShield telemetry with SIEM platforms including Splunk and Microsoft Sentinel.

Advanced Deepfake and Synthetic Media Detection: Integrating computer vision models capable of detecting AI-generated deepfake images and synthetic videos within web pages will address the emerging threat of AI-generated content used in social engineering attacks, visual phishing, and identity fraud.

Blockchain-Based Threat Intelligence Sharing: A decentralized threat intelligence protocol using distributed ledger technology would enable secure, tamper-proof sharing of novel threat indicators between WebFoxShield instances across organizations, accelerating community-driven detection of emerging attack patterns without centralized data aggregation.

Voice Phishing (Vishing) Detection: Extending detection to WebRTC audio streams would enable real-time identification of social engineering patterns during browser-based voice and video calls, addressing the growing vishing threat vector that currently falls entirely outside browser security tool coverage.

REFERENCES

1. S. Panja, S. Mondal, A. Nag, J. P. Singh, M. J. Saikia, and A. K. Barman, "An Efficient Malware Detection Approach Based on Machine Learning Feature Influence Techniques for Resource-Constrained Devices," IEEE Transactions on Dependable and Secure Computing, 2025.
2. M. A. Hossain and M. S. Islam, "Enhanced Detection of Obfuscated Malware in Memory Dumps: A Machine Learning Approach for Advanced Cybersecurity," Computers & Security, vol. 140, p. 103768, 2024.
3. M. Goyal and R. Kumar, "Machine Learning for Malware Detection on Balanced and Imbalanced Datasets," International Journal of Information Security, vol. 19, no. 4, pp. 479–491, 2020.
4. A. Kumar et al., "Malware Detection Using Machine Learning," in Proc. IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), 2020.
5. V. Patel, S. Choe, and T. Halabi, "Predicting Future Malware Attacks on Cloud Systems Using Machine Learning," IEEE Access, vol. 8, pp. 217627–217640, 2020.
6. M. Sabir, J. A. Shah, M. A. Khan, and F. Algarni, "Machine Learning Based Phishing Website Detection Using URL Features," IEEE Access, vol. 9, pp. 41966–41985, 2021.
7. A. Aljofey, Q. Jiang, Q. Qu, M. Huang, and J.-P. Niyigena, "An Effective Phishing Detection Model Based on Character-Level Convolutional Neural Network from URL," Electronics, vol. 10, no. 13, p. 1514, 2021.
8. Y. Zhang and X. Chen, "A Deep Learning Approach for Detecting Phishing Websites from Their Visual Appearance," Computers & Security, vol. 113, p. 102553, 2022.
9. A. K. Jain and B. B. Gupta, "Detection of Phishing Websites Using Hybrid Features," International Journal of Advanced Intelligence Paradigms, vol. 15, no. 2, pp. 181–200, 2020.
10. R. S. Rao, T. A. A. Vignesh, A. R. Pais, and G. Bhatt, "Real-Time Phishing URL Detection Using Machine Learning," ACM Digital Library, 2023.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802248

11. C.Nagarajan and M.Madheswaran - 'Stability Analysis of Series Parallel Resonant Converter with Fuzzy Logic Controller Using State Space Techniques'- Taylor &Francis, Electric Power Components and Systems, Vol.39 (8), pp.780-793, May 2011. DOI: 10.1080/15325008.2010.541746
12. C.Nagarajan and M.Madheswaran - 'Experimental verification and stability state space analysis of CLL-T Series Parallel Resonant Converter' - Journal of Electrical Engineering, Vol.63 (6), pp.365-372, Dec.2012. DOI: 10.2478/v10187-012-0054-2
13. C.Nagarajan and M.Madheswaran - 'Performance Analysis of LCL-T Resonant Converter with Fuzzy/PID Using State Space Analysis'- Springer, Electrical Engineering, Vol.93 (3), pp.167-178, September 2011. DOI 10.1007/s00202-011-0203-9
14. S.Tamilselvi, R.Prakash, C.Nagarajan,"Solar System Integrated Smart Grid Utilizing Hybrid Coot-Genetic Algorithm Optimized ANN Controller" Iranian Journal Of Science And Technology-Transactions Of Electrical Engineering, DOI10.1007/s40998-025-00917-z,2025
15. S.Tamilselvi, R.Prakash, C.Nagarajan," Adaptive sliding mode control of multilevel grid-connected inverters using reinforcement learning for enhanced LVRT performance" Electric Power Systems Research 253 (2026) 112428, doi.org/10.1016/j.epsr.2025.112428
16. S.Thirunavukkarasu, C. Nagarajan, 2024, "Performance Investigation on OCF and SCF study in BLDC machine using FTANN Controller," Journal of Electrical Engineering And Technology, Volume 20, pages 2675–2688, (2025), doi.org/10.1007/s42835-024-02126-w
17. C. Nagarajan, M.Madheswaran and D.Ramasubramanian- 'Development of DSP based Robust Control Method for General Resonant Converter Topologies using Transfer Function Model'- Acta Electrotechnica et Informatica Journal , Vol.13 (2), pp.18-31,April-June.2013, DOI: 10.2478/aeei-2013-0025.
18. C.Nagarajan and M.Madheswaran - 'DSP Based Fuzzy Controller for Series Parallel Resonant converter'- Springer, Frontiers of Electrical and Electronic Engineering, Vol. 7(4), pp. 438-446, Dec.12. DOI 10.1007/s11460-012-0212-0.
19. C.Nagarajan and M.Madheswaran - 'Experimental Study and steady state stability analysis of CLL-T Series Parallel Resonant Converter with Fuzzy controller using State Space Analysis'- Iranian Journal of Electrical & Electronic Engineering, Vol.8 (3), pp.259-267, September 2012.
20. C.Nagarajan and M.Madheswaran, "Analysis and Simulation of LCL Series Resonant Full Bridge Converter Using PWM Technique with Load Independent Operation" has been presented in ICTES'08, a IEEE / IET International Conference organized by M.G.R.University, Chennai.Vol.no.1, pp.190-195, Dec.2007
21. Suganthi Mullainathan, Ramesh Natarajan, "An SPSS and CNN modelling based quality assessment using ceramic materials and membrane filtration techniques", Revista Materia (Rio J.) Vol. 30, 2025, DOI: https://doi.org/10.1590/1517-7076-RMAT-2024-0721
22. M Suganthi, N Ramesh, "Treatment of water using natural zeolite as membrane filter", Journal of Environmental Protection and Ecology, Volume 23, Issue 2, pp: 520-530,2022
23. A. Singh and M. Kaur, "Sensitive Data Leakage Detection in Web Applications Using Pattern Matching and NLP," Journal of Web Engineering, vol. 21, no. 5, pp. 1345–1370, 2022.
24. J. Park, D. Kim, and H. Lee, "Preventing Data Leakage Using Client-Side Monitoring in Web Applications," International Journal of Web Services Research, vol. 18, no. 3, pp. 44–62, 2021.
25. P. Kumar and A. Mishra, "Browser Extension Based Detection of Phishing and Malicious Websites," Journal of Cybersecurity, vol. 9, no. 1, 2023.
26. A. Torres, J. Sanz-Cruzado, and C. Castillo, "AI-Based Detection of Malicious Web Pages Using Decision Trees and Random Forest," Expert Systems with Applications, vol. 192, p. 116409, 2022.
27. Anti-Phishing Working Group (APWG), "Phishing Activity Trends Report Q2 2024," [Online]. Available: https://apwg.org/trendsreports/ [Accessed: March 2026].
28. Have I Been Pwned, "HIBP v3 API Documentation," [Online]. Available: https://haveibeenpwned.com/API/v3 [Accessed: March 2026].
29. VirusTotal, "VirusTotal API v3 Documentation," [Online]. Available: https://developers.virustotal.com/reference/overview [Accessed: March 2026].
30. S. Roopak, G. Y. Tian, and J. Chambers, "Deep Learning Models for Cyber Security in IoT Networks," in Proc. IEEE Annual Computing and Communication Workshop and Conference (CCWC), 2019.
31. Kiran, A., Rubini, P., & Kumar, S. S. (2025). Comprehensive review of privacy, utility and fairness offered by synthetic data. IEEE Access.
32. Gopinathan, V. R. (2024). Real-Time Financial Risk Intelligence Using Secure-by-Design AI in SAP-Enabled Cloud Digital Banking. International Journal of Computer Technology and Electronics Communication, 7(6), 9837-9845.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802248

33. Udayakumar, R., Elankavi, R., Vimal, R., & Sugumar, R. (2023). Improved Particle Swarm Optimization with Deep Learning-Based Municipal Solid Waste Management in Smart Cities. Environmental & Social Management Journal, 17(4).
34. Anand, L. (2023). An Intelligent AI and ML–Driven Cloud Security Framework for Financial Workflows and Wastewater Analytics. International Journal of Humanities and Information Technology, 5(02), 87-94.
35. Soundappan, S. J. (2020). Big Data Analytics in Healthcare: Applications for Pandemic Forecasting. International Journal of Advanced Research in Computer Science & Technology, 3(1), 2248-2253.
36. Rajasekar, M. (2024). Real-Time Predictive DevOps Intelligence for Risk-Aware Digital Business Processes in Cloud and SAP Ecosystems. International Journal of Advanced Research in Computer Science & Technology, 7(4), 10713-10718.
37. Poornima, G., & Anand, L. (2024, May). Novel AI Multimodal Approach for Combating Against Pulmonary Carcinoma. In 2024 5th International Conference for Emerging Technology (INCET) (pp. 1-6). IEEE.
38. Prabha, P. S., & Rengarajan, A. (2025). Adaptive Cloud Resource Allocation Using Attention-Driven Deep Reinforcement Learning. Engineering, Technology & Applied Science Research, 15(6), 29334-29340.
39. Jagadeesh, S., & Sugumar, R. (2017). A Comparative study on Artificial Bee Colony with modified ABC algorithm. European Journal of Applied Sciences, 9(5), 243-248.
40. Varma, K. K., & Anand, L. (2025, March). Deep Learning Driven Proactive Auto Scaler for High-Quality Cloud Services. In International Conference on Computing and Communication Systems for Industrial Applications (pp. 329-338). Singapore: Springer Nature Singapore.
41. Kumar, S. A., & Anand, L. (2025). A Novel EEG-Based Deep Learning Framework for Enhancing Communication in Locked-In Syndrome Using P300 Speller and Attention Mechanisms. KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS, 19(11), 3841-3855.
42. Poornima, G., & Anand, L. (2025). Medical image fusion model using CT and MRI images based on dual scale weighted fusion based residual attention network with encoder-decoder architecture. Biomedical Signal Processing and Control, 108, 107932.
43. Archana, R., & Anand, L. (2025). Residual u-net with Self-Attention based deep convolutional adaptive capsule network for liver cancer segmentation and classification. Biomedical Signal Processing and Control, 105, 107665.Kumar, S. A., & Anand, L. (2025). A Novel EEG-Based Deep Learning Framework for Enhancing Communication in Locked-In Syndrome Using P300 Speller and Attention Mechanisms. KSII Transactions on Internet and Information Systems, 19(11), 3841-3855.
44. Rengarajan, A. (2025). Cloud-Based AI-Driven Threat Detection Framework for Smart Grid Cybersecurity. International Journal of Future Innovative Science and Technology, 8(6), 16065.
45. Murugeshwari, B., Sudharson, K., Panimalar, S. P., Shanmugapriya, M., & Abinaya, M. (2020). SAFE–Secure Authentication in Federated Environment using CEG Key code.
46. Raj A. A., & Sugumar, R. (2023). Early Detection of COVID-19 with Impact on Cardiovascular Complications using CNN Utilising Pre-Processed Chest X-Ray Images. 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC), IEEE.
47. Jagadeesh, S., & Sugumar, R. (2017). A Comparative study on Artificial Bee Colony with modified ABC algorithm. European Journal of Applied Sciences, 9(5), 243-248.
48. Selvi, G. V., Anbarasan, A. B., Murthy, B. A., & Prabavathy, S. (2023). An Application Oriented Integrated Unequal Clustering Algorithm for Wireless Sensor Network. In Underwater Vehicle Control and Communication Systems Based on Machine Learning Techniques (pp. 140-154). CRC Press.
49. Sruthi, R. S., Ananya, S., & Murugeshwari, B. (2010). Web Based Virtual Control System Laboratory and On-Line Temperature Control of Electrophoresis Equipment using LabVIEW. International Journal of Computer Applications, 975, 8887.
50. Vimal Raja, G. (2021). Mining Customer Sentiments from Financial Feedback and Reviews using Data Mining Algorithms. International Journal of Innovative Research in Computer and Communication Engineering, 9(12), 14705-14710.
51. MATHEW, A. R. (2025). Neurosecurity and Brain-Computer Interfaces.
52. Soundappan, S. J. (2024). AI-Driven Customer Intelligence in Enterprise Lakehouse Systems Sentiment Mining Governance-Aware Analytics and Real-Time Data Synchronization. International Journal of Advanced Engineering Science and Information Technology (IJAESIT), 7(5), 14905.
53. Mathew, A. (2025). Human–AI Collaboration in Security Operations: Measuring Alert Trust, Automation Bias, and Analyst Upskilling in AI-Augmented SOC Environments. International Journal of Computer Technology and Electronics Communication, 8(5), 11375-11380.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802248

54. Soundappan, S. J. (2022). AI-Based Fault Detection and Isolation for Reliability in Modern Power Systems. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 5(4), 7106-7110.
55. Poornima, G., & Anand, L. (2024, April). Effective Machine Learning Methods for the Detection of Pulmonary Carcinoma. In 2024 Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM) (pp. 1-7). IEEE.Garg, V. K., Soundappan, S. J., & Kaur, E. M. (2020). Enhancement in intrusion detection system for WLAN using genetic algorithms. South Asian Research Journal of Engineering and Technology, 2(6), 62–64.
56. Rengarajan, A., Jayakumar, C., & Sugumar, R. (2012). Optimization Of Recent Attacks Using Internet Protocol. National Journal of System and Information Technology, 5(1), 8.
57. Mathew, A. (2024). AI TRiSM: Trust, Risk, and Security Management in Cybersecurity. Cybersecurity, 4(3), 84-90.
58. Mathew, A. (2025). Deep seek vs. ChatGPT: A deep dive into AI Language mastery. Int J Multidisciplinary Res, 7(1), 1-5.